

06-05-00

A

01/2000 FORM 1

Sheet 1 of 4

TRANSMITTAL FORM FOR FILING PATENT APPLICATION

Attorney

Docket No.: SYNER-161XX

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
 Ten Post Office Square
 Boston, Massachusetts 02109
 Telephone: (617) 542-2290
 Telecopier: (617) 451-0313

Express Mail No: EL418425315US

BOX PATENT APPLICATION
 Assistant Commissioner for Patents
 Washington, D.C. 20231

Date: June 2, 2000

First Named Inventor or Application
 Identifier: Smaragda Hadjinikitas

Sir:

Transmitted herewith under 37 CFR § 1.53 for filing is the patent application of:

Inventor: Smaragda Hadjinikitas, Kenneth J. Blanc, Jeffrey R. Young

Entitled: DISTRIBUTED SYSTEM AUTHENTICATION

☐ This is a request for filing a ☐ **continuation** ☐ **divisional** ☐ **continuation**
in-part application under §1.53(b) of prior Application No. _____, filed
 _____ entitled:

Enclosed are:

☒ 33 pages of written description, claims and Abstract, inclusive☒ 4 sheets of ☐ informal ☒ formal drawings of Figs. 1-4 (one set.)☒ Oath or Declaration☒ Newly executed (original or copy)☐ Copy from prior application (37 CFR 1.63(d)) (for continuation/divisional).

The entire disclosure of the prior application, from which a copy of the oath
 or declaration is supplied, is considered as being part of the disclosure of
 the accompanying application and is hereby incorporated by reference therein.

☐ To be filed later☒ Cover sheet and Assignment of the invention to: 3Com Corporation

☐ Certified copy of a _____ application (if foreign priority is
 claimed) with letter claiming priority under Rule 55.

☐ Information Disclosure Statement with ___ citations☐ Preliminary amendment is enclosed.☒ Return receipt postcard☐ Other:

TRANSMITTAL FORM FOR FILING PATENT APPLICATION (CONTINUED)

Attorney
Docket No.: SYNER-161XX

- [] Verified statement of Small Entity status (\$1.9 and \$1.27)
- [] Verified statement of Small Entity was filed in prior application. Status still proper and desired
- [] Priority is claimed under 35 USC § 120 as indicated on the attached sheet 4.
- [] Priority is claimed under 35 USC §119(a)-(d) as indicated on the attached sheet 4.
- [] Priority is claimed under 35 USC §119 (e) as indicated on the attached sheet 4.

[X] Richard E. Gamache is hereby appointed Associate Attorney by:
Registration No.: 39,196

Victor B. Lebovici
Attorney of Record: Victor B. Lebovici
Registration No.: 30,864

[] **Power of Attorney** in the originally-filed application has been granted to one or more of the registered attorneys listed below. The attorneys listed below not previously granted power in the originally-filed application, as well as _____, are hereby given associate power:

Registration No.:

Stanley M. Schurgin, Reg. No. 20,979
Charles L. Gagnebin III, Reg. No. 25,467
Paul J. Hayes, Reg. No. 28,307
Victor B. Lebovici, Reg. No. 30,864

Eugene A. Feher, Reg. No. 33,171
Beverly E. Hjorth, Reg. No. 32,033
Holliday C. Heine, Reg. No. 34,346
Gordon R. Moriarty, Reg. No. 38,973

[] Cancel in this application original claims _____ of the prior application before calculating the filing fee.

[] Add in this application claims _____ per amendment before calculating fee.

CLAIMS FILED:	MINUS BASE:	EXTRA CLAIMS:	RATE:	BASIC FEE:
				\$690.00
Independent	6 - 3	= 3	x \$78.00 =	234.00
Total	11 - 20	=	x \$18.00 =	0.00
[] Multiple Dependent Claims (1st presentation)			+ \$260.00 =	0.00
SUBTOTAL FILING FEE				\$924.00
Small Entity filing, divide by 2. (Note: verified statement must be attached per \$1.9, \$1.27, \$1.28.)				0.00
TOTAL Filing Fee				\$924.00


Attorney Docket No.: SYNER-161XX

TRANSMITTAL FOR FILING PATENT APPLICATION (CONTINUED)

- [X] The filing fee has been calculated above; a check in the amount of \$924.00 is enclosed.
- [] The filing fee will be submitted at a later date.
- [X] In the event a Petition for Extension of Time under 37 CFR \$1.17 is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.
- [X] The Commissioner is hereby authorized to charge payment of any additional filing fees under 37 CFR \$1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.

Address all future communications to:

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313



Attorney of Record: Victor B. Lebovici
Registration No. 30,864

Attorney Docket No.: SYNER-161XX

TRANSMITTAL FOR FILING PATENT APPLICATION (CONTINUED)

☐ Priority is claimed under 35 USC § 120 of prior Application(s)
No. _____, filed _____, entitled:

☐ The above-identified application(s) is/are assigned of record to:

☐ Priority is claimed under 35 USC § 119 (a)-(d) of the following application(s).

(Application Number) (Country) (Filing Date)

(Application Number) (Country) (Filing Date)

(Application Number) (Country) (Filing Date)

☐ The above-identified application(s) is/are assigned of record to:

☐ Priority is claimed under 35 USC § 119 (e) of the following provisional application(s).

(Application Number) (Filing Date)

(Application Number) (Filing Date)

(Application Number) (Filing Date)

☐ The above-identified provisional application(s) is/are assigned of record to:

☐ The claim of small entity status in the above-identified provisional application(s) is made in this application and a copy of the small entity form(s) from the provisional application(s) is/are enclosed.

SUBMIT IN TRIPLICATE

226342

5

TITLE OF THE INVENTION
DISTRIBUTED SYSTEM AUTHENTICATION

CROSS REFERENCE TO RELATED APPLICATIONS

N/A

10

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR
DEVELOPMENT

N/A

15

BACKGROUND OF THE INVENTION

The present invention relates to systems and techniques for authenticating users submitting requests from client computers for services/resources provided by server computers executing distributed applications.

20

In a typical computer network configuration, client computers interconnected by the computer network transmit user requests to access services/resources provided by server computers connected to the network. Such server computers typically include data processing agents, which execute applications for processing the user requests and providing the requested services/resources to the client computers. These applications may be executed on a single data processing agent to provide at least one service/resource to the client computers. Alternatively, these applications may be distributed such that portions of

25

30

ATTORNEY DOCKET NO. SYNER-161XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

Express Mail Number

EL41842531505

000000 24258560

the distributed application are executed on respective data processing agents included in the server computer. As a result, each respective data processing agent may be used for providing specific services/resources to the client computers.

One drawback of server computers, whether they execute distributed or non-distributed applications, is that they typically have no knowledge about the user's right to access requested services/resources. This can be problematic because applications executing on server computers often provide different services/resources depending upon the user's level of access privileges.

For example, a user of a client computer with a particular level of access privileges may or may not have access to, e.g., specific files, directories, databases, web pages, and/or other computer services/resources provided by the application. It is therefore desirable to authenticate users submitting requests from client computers to ensure that they have the requisite levels of access privileges for accessing the requested files, directories, databases, web pages, and/or other computer services/resources. In this way, unauthorized users can be prevented from accessing restricted services/resources on the computer network, and the security of the computer network can be maintained.

One technique for authenticating users includes receiving a user request from a client computer at a server computer for a service/resource provided by an application resident on the server computer; and, in response to that request, transmitting a message from the server computer to the client computer informing the client computer of what it

must do to authenticate the user. For example, that message might inform the client computer that in order to authenticate the user it must provide a valid USERNAME/PASSWORD combination. In response to that message, the user enters the required USERNAME/PASSWORD combination at the client computer. Another user request is then received at the server computer from the client computer including the entered USERNAME/PASSWORD combination. In response to that request, the USERNAME is located in, e.g., a stored access control list; the PASSWORD corresponding to the USERNAME is verified; and, if the USERNAME/PASSWORD combination is found valid, a stored level of access privileges is retrieved for that user. Finally, the application executing on the server computer provides the user of the client computer with the requested services/resources according to that user's level of access privileges.

The above-described technique of authenticating users can be implemented on a server computer with a single data processing agent executing a non-distributed application that requires knowledge of the user's access privilege level. However, this technique has drawbacks when implemented on a server computer with a plurality of data processing agents executing a distributed application because it has no mechanism for providing the user's access privilege level to the application executing on the plurality of agents.

It would therefore be desirable to have a system and technique for authenticating users submitting requests from client computers to a server computer executing a

distributed application. It would also be desirable to have such systems and techniques for authenticating users that minimize the overall time required for performing user authentication.

5

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a method and apparatus are disclosed for authenticating a user submitting a service request from a client computer to a server computer executing a distributed application on a plurality of data processing agents. Such user authentication is accomplished by providing a centralized mechanism that all data processing agents of the server can utilize to authenticate a potential user.

In one embodiment, a first data processing agent included in the server receives a service request from a potential user, and submits an authentication request to a second data processing agent included in the server to authenticate the user. The second data processing agent attempts to authenticate the user, and transmits a message to the first data processing agent including information indicative of whether the user is successfully authenticated. If the user is successfully authenticated, then the first data processing agent shares that information with the distributed application executing thereon, which provides the requested service to the user. In the foregoing manner, the second data processing agent serves as the centralized mechanism that the first data processing agent and all of the remaining data processing agents

included in the server can utilize to authenticate potential users.

003090 " 24258520
09585747 060200

5 In a second embodiment, the first data processing agent included in the server receives a first service request from the user, and submits an authentication request to the second data processing agent to authenticate the user. The second data processing agent attempts to authenticate the user; and, if the user is successfully authenticated, stores a time-out value indicative of a predetermined time period for that user. Next, the second data processing agent determines whether the predetermined time period is exceeded starting from a time of receipt of the first request. In the event that the predetermined time period is exceeded without receiving a second service request from the user, 15 the server requires the user to be re-authenticated at the second data processing agent upon receipt of the second service request. In the foregoing manner, the second data processing agent restricts the amount of time that the user can remain idle before the server requires re-authentication 20 of that user.

25 In a third embodiment, the first data processing agent receives a service request from the user, and submits an authentication request to the second data processing agent to authenticate the user. Next, the second data processing agent attempts to authenticate the user. In the event that the user is successfully authenticated, the second data processing agent transmits valid user authentication information to the first data processing agent, which locally stores that information. Next, the first data 30 processing agent receives another service request from the

user, and attempts to authenticate the user using the stored user authentication information. If the user is successfully authenticated, then the first data processing agent shares that information with the distributed application executing thereon, which provides the requested service to the user. In the foregoing manner, the first data processing agent can authenticate the user without having to submit an authentication request to the second data processing agent.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

The invention will be more fully understood by reference to the following Detailed Description of the Invention in conjunction with the Drawing of which:

Fig. 1 is a block diagram illustrating a computer network operative in a manner according to the present invention;

Fig. 2 is a block diagram illustrating a representative server computer connected to the computer network depicted in Fig. 1, operative in a manner according to the present invention;

Fig. 3 is a flow diagram illustrating a method of the representative server computer depicted in Fig. 2 for authenticating users, operative in a manner according to the present invention; and

Fig. 4 is a flow diagram illustrating a method of the representative server computer depicted in Fig. 2 for restricting the amount of time that a valid user can remain idle, operative in a manner according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 depicts an illustrative embodiment of a computer network 100 that is operative in a manner in accordance with the present invention. Specifically, the computer network 100 includes a plurality of client computers ("clients") such as clients 102, 104, and 106, and at least one server computer ("server") such as server 108. Further, the clients 102, 104, and 106, and the server 108 are operatively connected to a network 110, which may comprise a Local Area Network (LAN), a Wide Area Network (WAN), the Internet, or any other network suitable for linking clients and servers to allow communications therebetween.

Each of the clients 102, 104, and 106, and the server 108, includes a network adapter (not shown) for enabling communications over the network 110. In addition, each client 102, 104, and 106 includes at least one memory (not shown) such as a ROM or RAM, and at least one processor (not shown) operative for executing programs stored in the memory, including applications for processing user inputs, initiating and/or controlling connections to the network 110, and submitting requests for services/resources to the server 108.

For example, users (not shown) of the clients 102, 104, and 106 may submit requests to the server 108 for accessing selected files, directories, databases, web pages, and/or other computer services/resources. Those of ordinary skill in this art will recognize that the term "users" may refer not only to human operators, but also to processes executing on the clients 102, 104, and 106.

Fig. 2 depicts a block diagram of an exemplary server 108 such as depicted in Fig. 1. The server 108 includes a manager agent 202 operatively connected to a plurality of service agents, such as service agents 204, 206, and 208, via a buss 232. Each agent 202, 204, 206, and 208 includes at least one memory (not shown) such as a ROM or RAM, and at least one processor (not shown) operative for executing programs stored in the memory, including applications for initiating and/or controlling connections to the network 110, processing requests for services/resources submitted by the clients 102, 104, and 106, and providing requested services/resources to the clients 102, 104, and 106.

In this illustrative embodiment, the manager agent 202 and the service agents 204, 206, and 208 are capable of executing distributed applications. For example, the users of the clients 102, 104, and 106 may submit requests to the server 108 for accessing services/resources provided by a distributed application executing on the plurality of agents 202, 204, 206, and 208. Further, each agent 202, 204, 206, and 208 executing the distributed application may provide specific services/resources to the clients 102, 104, and 106 based on the requests submitted by the users.

It should be noted that the specific services/resources provided by way of the agents 202, 204, 206, and 208 to the clients 102, 104, and 106 are dependent upon each user's level of access privileges, which is determined at the server 108 through the execution of a user authentication program. Because each agent 202, 204, 206, and 208 executing the distributed application provides specific services/resources to the users, each agent 202, 204, 206,

and 208 uses at least a portion of its processing and storage resources for performing tasks related to user authentication.

Specifically, each agent 202, 204, 206, and 208
5 includes an authentication client operatively connected to a writable, local storage media such as a RAM. For example, the manager agent 202 includes an authentication client 214 connected to a local storage media 224, the service agent 204 includes an authentication client 216 connected to a
10 local storage media 226, the service agent 206 includes an authentication client 218 connected to a local storage media 228, and the service agent 208 includes an authentication client 220 connected to a local storage media 230. The manager agent 202 further includes an authentication server
15 212 connected to a writable, local storage media 223 such as a RAM; and, a writable, main storage media 222 such as a non-volatile RAM.

In the illustrative embodiment, upon receipt of a user request for a specific service/resource provided by a
20 distributed application executing on one or more of the agents 202, 204, 206, and 208, the authentication client for a respective agent attempts to authenticate the user by accessing user data stored in the local storage media connected thereto. If the authentication client
25 successfully locates authentication information for the user in the local storage media, but that information does not match corresponding authentication information attached to the user request, then the authentication client transmits a message to the user informing him or her that access to the
30 requested service/resource is denied. Alternatively, if the

authentication client cannot locate authentication information for the user in the local storage media, then the authentication client submits a request to the authentication server 212 to authenticate the user.

5 In response to that request, the authentication server 212 attempts to authenticate the user by accessing user data stored in the local storage media 223 connected thereto. If the authentication server successfully locates authentication information for the user in the local storage
10 media 223, but that information does not match the corresponding authentication information attached to the user request, then the authentication server transmits a message to the user informing him or her that access to the requested service/resource is denied. Alternatively, if the
15 authentication server cannot locate authentication information for the user in the local storage media 223, then the authentication server 212 attempts to authenticate the user by accessing user data stored in the main storage media 222 connected thereto. If the authentication server
20 212 is incapable of successfully authenticating the user by accessing the user data stored in the main storage media 222, then the authentication server 212 may transmit a message to the user informing him or her that access to the requested service/resource is denied. Alternatively, the
25 authentication server 212 may transmit a message to the user prompting him or her to "log-in" by entering valid user authentication information. Upon receipt of that information, the authentication server 212 attempts to authenticate the user by comparing the entered user
30 authentication information with user data stored in the main

storage media 222. If the entered user authentication information matches corresponding user data stored in the main storage media 222, then access to the requested service/resource is permitted; otherwise, access is denied.

5 The illustrative embodiment disclosed herein will be better understood with reference to the following example, wherein a user, i.e., a human operator (not shown), of the client 102 wishes to obtain access to a specific service/resource provided by a distributed application
10 executing on the agents 202, 204, 206, and 208 of the server 108 (see Fig. 1).

Accordingly, the client 102 transmits a message to the server 108 including a request to access the specific service/resource, which is provided by the distributed
15 application executing on one of the agents 202, 204, 206, and 208; for example, the service agent 204. It should be understood that the manner in which the client 102 and the server 108 transmit and receive messages is conventional.

In this illustrative example, it is assumed that the
20 above-mentioned request for services/resources transmitted by the client 102 is the first request of a session, and the first request does not include information for authenticating the user of the client 102. Accordingly, in response to that first request, the server 108 transmits a
25 message to the client 102 that includes information about what the client 102 must do to authenticate the user.

Specifically, that message informs the client 102 that it must provide the authentication server 212 with valid authentication information for the user, e.g., a valid
30 USERNAME/PASSWORD combination. The client 102 therefore

prompts the user to enter the required USERNAME/PASSWORD combination. For example, the client 102 may prompt the user by way of a user interface (not shown), which includes a display monitor, and a keyboard and/or a screen-cursor manipulator such as a mouse.

After the user enters the requested USERNAME/PASSWORD combination via the user interface, the client 102 transmits another request for services/resources to the server 108 along with the entered USERNAME/PASSWORD. In response to that request, the authentication server 212 verifies the entered USERNAME/PASSWORD combination against user data stored in the main storage media 222 included in the manager agent 202.

In this illustrative example, the main storage media 222 includes user data corresponding to a list of "permissible" users; i.e., users that would be permitted access to the services/resources provided by the distributed application executing on the server 108 upon verification of a valid USERNAME/PASSWORD combination. For example, user data corresponding to the permissible users may be arranged in the main storage media 222 as a MAIN USER LOGIN TABLE, including the USERNAME for each permissible user and corresponding PASSWORD and ACCESS LEVEL for that user. Further, the ACCESS LEVEL may be indicated by, e.g., a numerical value within a specified range of numerical values, with each numerical value indicating a different level of access privileges for the user and optionally including the default value, ACCESS DENIED, indicating that access to the requested service/resource is denied.

002090" 24258560

If the authentication server 212 (1) locates the entered USERNAME in the MAIN USER LOGIN TABLE, and (2) determines that the entered PASSWORD matches the corresponding PASSWORD in the MAIN USER LOGIN TABLE, then the entered USERNAME/PASSWORD combination is verified and the user of the client 102 is successfully authenticated. As a result, the authentication server 212 retrieves the ACCESS LEVEL for that user from the MAIN USER LOGIN TABLE stored in the main storage media 222; stores the USERNAME, PASSWORD, and ACCESS LEVEL information in the local storage media 223; and, transmits the USERNAME, PASSWORD, and ACCESS LEVEL information to the authentication client requesting authentication of the user (e.g., the authentication client 216) for storage in the local storage media operatively connected thereto (e.g., the local storage media 226).

In this illustrative example, each local storage media 223, 224, 226, 228, and 230 includes user data related to a list of "active" users; i.e., users that have been successfully verified against the user data stored in the main storage media 222. For example, the user data corresponding to the active users may be arranged in each local storage media 223, 224, 226, 228, and 230 as a LOCAL USER LOGIN TABLE, including the USERNAME for each active user and the corresponding PASSWORD and ACCESS LEVEL for that user.

The authentication server 212 also retrieves a SYSTEM TIMEOUT VALUE from the main storage media 222 along with the ACCESS LEVEL information, and transmits the SYSTEM TIMEOUT VALUE with the USERNAME, PASSWORD, and ACCESS LEVEL information to the authentication client requesting

In this illustrative example, the user may re-configure his or her corresponding LOGIN TIMEOUT VALUE via a user interface of the server 108. For example, the SYSTEM TIMEOUT VALUE may be equal to 30 minutes; and, the user may re-configure the corresponding LOGIN TIMEOUT VALUE to equal any integral value ranging from 0 to 30 minutes, wherein a LOGIN TIMEOUT VALUE of 0 minutes indicates that the user is not subject to any time constraints between successive user activities and is therefore logged-in indefinitely.

ATTORNEY DOCKET NO. SYNER-161XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

In a preferred embodiment, the LOGIN TIMEOUT VALUES for active users are arranged in the local storage media 223 as a SESSION TIMEOUT TABLE, including the USERNAME for each active user and the corresponding LOGIN TIMEOUT VALUE for that user. Specifically, the SESSION TIMEOUT TABLE tracks LOGIN TIMEOUT VALUES for all active users of clients requesting services/resources provided by the distributed application executing on the agents 202, 204, 206, and 208 of the server 108.

After the authentication server 212 transmits the USERNAME, PASSWORD, ACCESS LEVEL, and SYSTEM TIMEOUT VALUE for the authenticated user to the authentication clients 214, 216, 218, and 220 for storage in the LOCAL USER LOGIN TABLES of the local storage media 224, 226, 228, and 230, respectively, the authentication server 212 provides the authenticated user's ACCESS LEVEL to the distributed application, which provides the requested service/resource to the user of the client 102 according to the ACCESS LEVEL of that user.

It should be noted that if the authentication server 212 fails to locate the entered USERNAME in the MAIN USER LOGIN TABLE, then that USERNAME does not correspond to a permissible user, and that user therefore cannot be successfully authenticated. As a result, the authentication server 212 transmits a message to the client 102 indicating that access to the requested service/resources is denied.

The illustrative embodiment disclosed herein can be used with distributed applications that require "per-form" authentication. "Per-form" authentication pertains to an authentication technique that requires verification of user

002090 24258560

authentication information for each form or document requested by the user. Because such user authentication information is stored locally in each of the agents 202, 204, 206, and 208, verification of user authentication information per-form may be achieved by locating that information in the LOCAL USER LOGIN TABLES of the local storage media 224, 226, 228, and 230, instead of submitting a request to the authentication server 212 to authenticate the user by locating that information in either the LOCAL USER LOGIN TABLE of the local storage media 223 or the MAIN USER LOGIN TABLE of the main storage media 222.

In this illustrative example, the user submits a second request for services/resources provided by the distributed application executing on the service agent 204. Because that second request is submitted after the authentication server 212 has already successfully authenticated the user, the client 102 automatically attaches the user's authentication information, i.e., the valid USERNAME/PASSWORD combination, to the request.

In response to that request, the authentication client 216 included in the service agent 204 attempts to authenticate the user by verifying the USERNAME/PASSWORD combination attached to the request against the user data stored in the local storage media 226 included in the service agent 204. If the authentication client 216 (1) locates the attached USERNAME in the LOCAL USER LOGIN TABLE of the local storage media 226, and (2) determines that the attached PASSWORD matches the corresponding PASSWORD in the LOCAL USER LOGIN TABLE, then the attached USERNAME/PASSWORD combination is verified and the user is successfully

authenticated. Accordingly, the authentication client 216 provides the authenticated user's ACCESS LEVEL to the distributed application, which provides the requested service/resource to the user according to the ACCESS LEVEL
5 of that user.

If the authentication client 216 determines that the attached PASSWORD does not match the corresponding PASSWORD in the LOCAL USER LOGIN TABLE of the local storage media 226, then the authentication client 216 transmits a message
10 to the user informing him or her that access to the requested service/resource is denied. Alternatively, if the authentication client 216 cannot locate the attached USERNAME in the LOCAL USER LOGIN TABLE, then the authentication client 216 submits a request to the
15 authentication server 212 to authenticate the user of the client 102.

In response to that request, the authentication server 212 attempts to authenticate the user by verifying the USERNAME/PASSWORD combination attached to the request
20 against the user data stored in the local storage media 223 included in the manager agent 202. If the authentication server 212 (1) locates the attached USERNAME in the LOCAL USER LOGIN TABLE of the local storage media 223, and (2) determines that the attached PASSWORD matches the
25 corresponding PASSWORD in the LOCAL USER LOGIN TABLE, then the attached USERNAME/PASSWORD combination is verified and the user of the client 102 is successfully authenticated. Accordingly, the authentication server 212 provides the authenticated user's ACCESS LEVEL to the distributed

application, which provides the requested service/resource to the user according to the ACCESS LEVEL of that user.

5 If the authentication server 212 determines that the attached PASSWORD does not match the corresponding PASSWORD in the LOCAL USER LOGIN TABLE of the local storage media 223, then the authentication server 212 transmits a message to the user informing him or her that access to the requested service/resource is denied. Alternatively, if the authentication server 212 cannot locate the attached
10 USERNAME in the LOCAL USER LOGIN TABLE, then the authentication server 212 attempts to authenticate the user by verifying the USERNAME/PASSWORD combination against the user data stored in the main storage media 222 included in the manager agent 202. If the authentication server 212 (1)
15 locates the attached USERNAME in the MAIN USER LOGIN TABLE of the main storage media 222, and (2) determines that the attached PASSWORD matches the corresponding PASSWORD in the MAIN USER LOGIN TABLE, then the attached USERNAME/PASSWORD combination is verified and the user of the client 102 is
20 successfully authenticated. Accordingly, the authentication server 212 provides the authenticated user's ACCESS LEVEL to the distributed application, which provides the requested service/resource to the user according to the ACCESS LEVEL of that user.

25 If the authentication server 212 either cannot locate the attached USERNAME in the MAIN USER LOGIN TABLE of the main storage media 222, or determines that the attached PASSWORD does not match the corresponding PASSWORD in the MAIN USER LOGIN TABLE, then the authentication server 212
30 cannot successfully authenticate that user. Accordingly,

the authentication server 212 transmits a message to the client 102 indicating that access to the requested service/resources is denied, thereby terminating the current session between the client 102 and the server 108.

5 As mentioned above, the SESSION TIMEOUT TABLE included in the local storage media 223 tracks LOGIN TIMEOUT VALUES for all active users of clients submitting requests for services/resources provided by the distributed application executing on the server 108. Further, each LOGIN TIMEOUT
10 VALUE indicates the maximum allowable amount of time between successive user activities; i.e., the maximum allowable "idle" time for that user. Accordingly, the authentication server 212 includes a timer (not shown) for determining whether maximum allowable idle times corresponding to users
15 listed in the SESSION TIMEOUT TABLE have been exceeded.

Similarly, each authentication client 214, 216, 218, and 220 includes a timer (not shown) for determining whether the maximum allowable idle time, as indicated by the LOGIN TIMEOUT VALUES stored in the respective local storage media
20 224, 226, 228, and 230, has been exceeded for each active user listed in the respective LOCAL USER LOGIN TABLES.

If the authentication server 212 determines that the maximum allowable idle time, as indicated by the user's LOGIN TIMEOUT VALUE stored in the SESSION TIMEOUT TABLE, has
25 been exceeded, then that user is no longer considered an active user and the authentication server 212 re-sets the ACCESS LEVEL in the SESSION TIMEOUT TABLE for that previously active, authenticated user to the default value, ACCESS DENIED.

Accordingly, if the authentication server 212 subsequently attempts to authenticate the user by verifying the user authentication information against the user data listed in the LOCAL USER LOGIN TABLE of the local storage media 223 or the MAIN USER LOGIN TABLE of the main storage media 222, and determines from the SESSION TIMEOUT TABLE that the ACCESS LEVEL for that user is set to the default value, then the authentication server 212 removes the user authentication data for that user from the LOCAL USER LOGIN TABLE and the SESSION TIMEOUT TABLE of the local storage media 223 to terminate the current session, and the server 108 transmits a message to the client prompting the user to log-in by entering valid user authentication information, thereby starting a new session. It should be noted that the SESSION TIMEOUT TABLE may alternatively be implemented as fields in the LOCAL USER LOGIN TABLE of the local storage media 223.

If any authentication client 214, 216, 218, or 220 determines that the maximum allowable idle time, as indicated by the user's LOGIN TIMEOUT VALUE stored in the LOCAL USER LOGIN TABLES of the local storage media 224, 226, 228, and 230, has been exceeded, then the authentication client 214, 216, 218, or 220 immediately removes the authentication information for that user from its respective LOCAL USER LOGIN TABLE. Accordingly, upon receiving a subsequent request for services/resources from the user, the authentication client 214, 216, 218, or 220 submits a request to the authentication server 212 to authenticate the user.

Whenever any authentication client 214, 216, 218, and 220 processes a user activity, then that authentication client re-starts its determination of whether the maximum allowable idle time, as indicated by that user's LOGIN
5 TIMEOUT VALUE, has been exceeded. Further, the authentication client transmits a message to the authentication server 212 notifying the authentication server 212 that the user activity has occurred. As a result, the authentication server 212 re-starts its
10 determination of whether the maximum allowable idle time for that user has been exceeded. In this way, "time-out" determinations performed by the authentication clients 214, 216, 218, and 220 are synchronized with the time-out determinations performed by the authentication server 212.

15 In a preferred embodiment, each authentication client 214, 216, 218, and 220 transmits one notification message to the authentication server 212 for each user that is active during the preceding 60 seconds. Further, in order to avoid potential race conditions between the time-out
20 determinations performed by the authentication server 212 and the authentication clients 214, 216, 218, and 220, the authentication clients 214, 216, 218, and 220 adjust the LOGIN TIMEOUT VALUES listed in the LOCAL USER LOGIN TABLES to be approximately 1 minute less than the corresponding
25 LOGIN TIMEOUT VALUES listed in the SESSION TIMEOUT TABLE. As a result, users' maximum allowable idle times will be exceeded at the authentication clients 214, 216, 218, and 220 about 1 minute before they are exceeded at the authentication server 212, thereby ensuring that all final
30 time-out determinations are made by the authentication

server 212 using the SESSION TIMEOUT TABLE. It should be understood that the 1 minute time may be varied recognizing the objective that race conditions be eliminated.

002090" 24258560
5 A method of authenticating a user submitting a service request from a client to a server executing a distributed application on a plurality of data processing agents is illustrated by reference to Fig. 3. As depicted in step 302, a first user request is received at a service agent for a service/resource provided by the distributed application
10 executing on the service agent. Next, the service agent submits, as depicted in step 304, a request to the manager agent to authenticate the user. As depicted in step 306, the manager agent receives the authentication request and attempts to authenticate the user. Next, a decision is
15 made, as depicted in step 308, as to whether the user is successfully authenticated. If so, then the manager agent retrieves, as depicted in step 310, valid authentication information for that user and transmits, as depicted in step 312, that information to the service agent. Otherwise, the
20 server transmits, as depicted in step 314, a message to the client indicating that access to the requested service/resource is denied. As depicted in step 316, the service agent receives and stores the valid authentication information. Next, the distributed application executing on
25 the service agent provides, as depicted in step 318, the requested service/resource to the user. As depicted in step 320, a second user request is received at the same service agent. Next, the service agent attempts, as depicted in step 322, to authenticate the user using the stored user
30 authentication information. As depicted in step 324, a

002090 24253560

decision is made as to whether authentication information attached to the second user request matches the stored user authentication information. If so, then the distributed application executing on the service agent provides, as depicted in step 326, the requested service/resource to the user. Otherwise, the server transmits, as depicted in step 328, a message to the client indicating that access to the requested service/resource is denied.

A method of restricting the amount of time that a user submitting requests from a client can remain idle before requiring that user to be re-authenticated is illustrated by reference to Fig. 4. As depicted in step 402, a first user request is received at a service agent for a service/resource provided by the distributed application executing on the service agent. Next, the service agent submits, as depicted in step 404, a request to a manager agent to authenticate the user. As depicted in step 406, the manager agent receives the authentication request and attempts to authenticate the user. Next, a decision is made, as depicted in step 408, as to whether the user is successfully authenticated. If so, then the manager agent retrieves, as depicted in step 410, valid user authentication information; stores, as depicted in step 412, a predetermined time-out value for that user; and transmits, as depicted in step 414, the authentication information and the time-out value to the service agent. Otherwise, the server transmits, as depicted in step 416, a message to the client indicating that access to the requested service/resource is denied. As depicted in step 418, the service agent receives and stores the valid user

5

15

20

25

30

techniques for authenticating users can be implemented in a fully distributed computing environment.

Those of ordinary skill in the art will appreciate that computer programs for performing the presently described functions can be delivered to the server 108 in many forms including, but not limited to: (a) information permanently stored on non-writable storage media (e.g., read-only memory devices within a computer such as ROM or CD-ROM disks readable by a computer I/O attachment; (b) information alterably stored on writable storage media (e.g., floppy disks, tapes, read/write optical media and hard drives); or, (c) information conveyed to a computer through a communication media, for example, using base-band signaling or broadband signaling techniques, such as over computer or telephone networks via a modem.

In addition, while in this illustrative embodiment the functions are illustrated as being software-driven and executable out of memories by processors in the server 108, the presently described functions may alternatively be embodied in part or in whole using hardware components such as custom or semi-custom integrated circuits including Application Specific Integrated Circuits (ASICs), Programmable Logic Arrays (PLAs), state machines, controllers or other hardware components or devices, or a combination of hardware components and software.

Those of ordinary skill in the art should further appreciate that variations to and modification of the above-described systems and techniques for authenticating users submitting requests from clients to servers may be made without departing from the inventive concepts disclosed

herein. Accordingly, the present invention should be viewed as limited solely by the scope and spirit of the appended claims.

002090" 24253560

CLAIMS

What is claimed is:

1. A method of authenticating a user of a client computer
5 at a server computer executing a distributed application on
a plurality of data processing agents, comprising the steps
of:

receiving a service request from the user at a first
data processing agent;

10 submitting an authentication request from the first
data processing agent to a second data processing agent to
authenticate the user;

receiving a response to the authentication request at
the first data processing agent from the second data
15 processing agent; and

if the received response indicates that the user is
successfully authenticated, providing the requested service
to the user.

20 2. The method of claim 1, wherein the received response
includes a level of access privileges for the user, and the
providing step includes the step of determining the service
provided to the user based upon the user's access privilege
level.

25 3. The method of claim 1, further including the steps of
receiving the service request from the user at the first
data processing agent included in a first server, and
submitting the authentication request from the first data

processing agent to the second data processing agent included in a second server.

4. A system for authenticating a user of a client computer at a server computer executing a distributed application on a plurality of data processing agents, the system comprising:

a server including a first data processing agent for receiving a service request from the user and a second data processing agent for authenticating the user,

wherein the first data processing agent includes resources for submitting an authentication request to the second data processing agent to authenticate the user, and

wherein the second data processing agent includes resources for receiving the authentication request, attempting to authenticate the user, and transmitting a response indicative of whether the user is successfully authenticated to the first data processing agent.

5. A method of authenticating a user of a client computer at a server computer executing a distributed application on a plurality of data processing agents, comprising the steps of:

receiving a first service request from the user at a first data processing agent;

submitting an authentication request from the first data processing agent to a second data processing agent to authenticate the user;

authenticating the user at the second data processing agent;

if the user is successfully authenticated, storing a timeout value indicative of a predetermined time period;

determining whether the predetermined time period is exceeded starting from a time of receipt of the first
5 service request; and

if the predetermined time period is exceeded without receiving a second service request from the user, requiring the user to be authenticated at the second data processing agent upon receipt of the second service request.
10

6. The method of claim 5, further including the steps of receiving the second service request from the user, and determining whether the predetermined time period is exceeded starting from the time of receipt of the second
15 service request.

7. The method of claim 5, further including the steps of receiving the second service request from the user at the first data processing agent, transmitting a message from the
20 first data processing agent to the second data processing agent including a notification that the second service request is received, receiving the notification at the second data processing agent, and determining whether the predetermined time period is exceeded starting from the time
25 of receipt of the notification.

8. A system for authenticating a user of a client computer at a server computer executing a distributed application on a plurality of data processing agents, the system
30 comprising:

wherein the first data processing agent includes
5 resources for submitting an authentication request to the
second data processing agent to authenticate the user,

15 wherein the first data processing agent further
includes resources for requiring the user to be re-
authenticated at the second data processing agent upon
receipt of a second service request if the predetermined
time period is exceeded before the second service request is
20 received.

receiving a service request from the user at a first data processing agent;

ATTORNEY DOCKET NO. SYNER-161XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

authenticating the user at the second data processing agent;

if the user is successfully authenticated at the second data processing agent, storing user authentication information at the first data processing agent;

receiving a next service request from the user at the first data processing agent;

authenticating the user at the first data processing agent using the stored information; and

if the user is successfully authenticated at the first data processing agent, providing the requested service to the user.

10. The method of claim 9, further including the step of, if the user is not successfully authenticated at the first data processing agent, submitting an authentication request from the first data processing agent to the second data processing agent to authenticate the user.

11. A system for authenticating a user of a client computer at a server computer executing a distributed application on a plurality of data processing agents, the system comprising:

a server including a first data processing agent for receiving a service request and a second data processing agent for authenticating the user,

wherein the first data processing agent includes resources for submitting an authentication request to the second data processing agent to authenticate the user, storing user authentication information if the user is

successfully authenticated, receiving a next service request from the user, authenticating the user using the stored information, and providing the requested service to the user if the user is successfully authenticated.

002090 7458560

ABSTRACT OF THE DISCLOSURE

5 A system and technique for authenticating a user
submitting a service request from a client computer to a
server computer executing a distributed application on a
10 plurality of data processing agents. The server computer
includes a plurality of data processing agents including a
plurality of service agents and a manager agent, which
manages tasks related to user authentication. A first
service agent receives a service request from a user, and
15 submits an authentication request to the manager agent to
authenticate the user. In the event that the manager agent
successfully authenticates the user, the manager agent
transmits a message to the first service agent including
valid user authentication information, and stores a time-out
20 value indicative of a predetermined time period for that
user. The first service agent receives the valid user
authentication information, stores it, and shares it with
the distributed application executing thereon, which
provides the requested service to the user. The manager
25 agent determines whether the predetermined time period has
been exceeded starting from the time of receipt of the
service request. If that time period has not been exceeded
before another service request is received at the first
service agent, then the first service agent attempts to
30 authenticate the user using the stored authentication
information. Otherwise, the first service agent submits
another authentication request to the manager agent to
authenticate the user.

224613-1

ATTORNEY DOCKET NO. SYNER-161XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

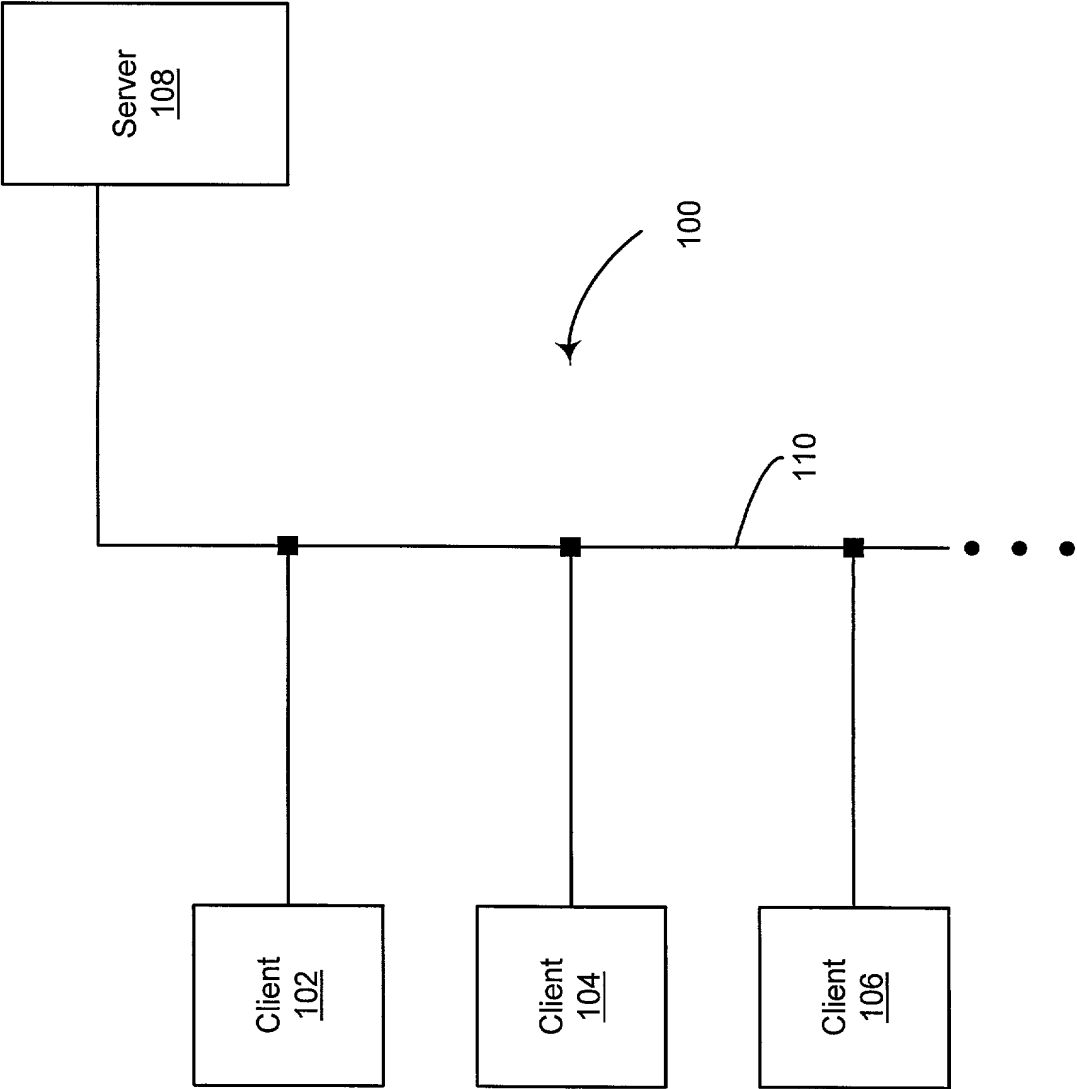


Fig. 1

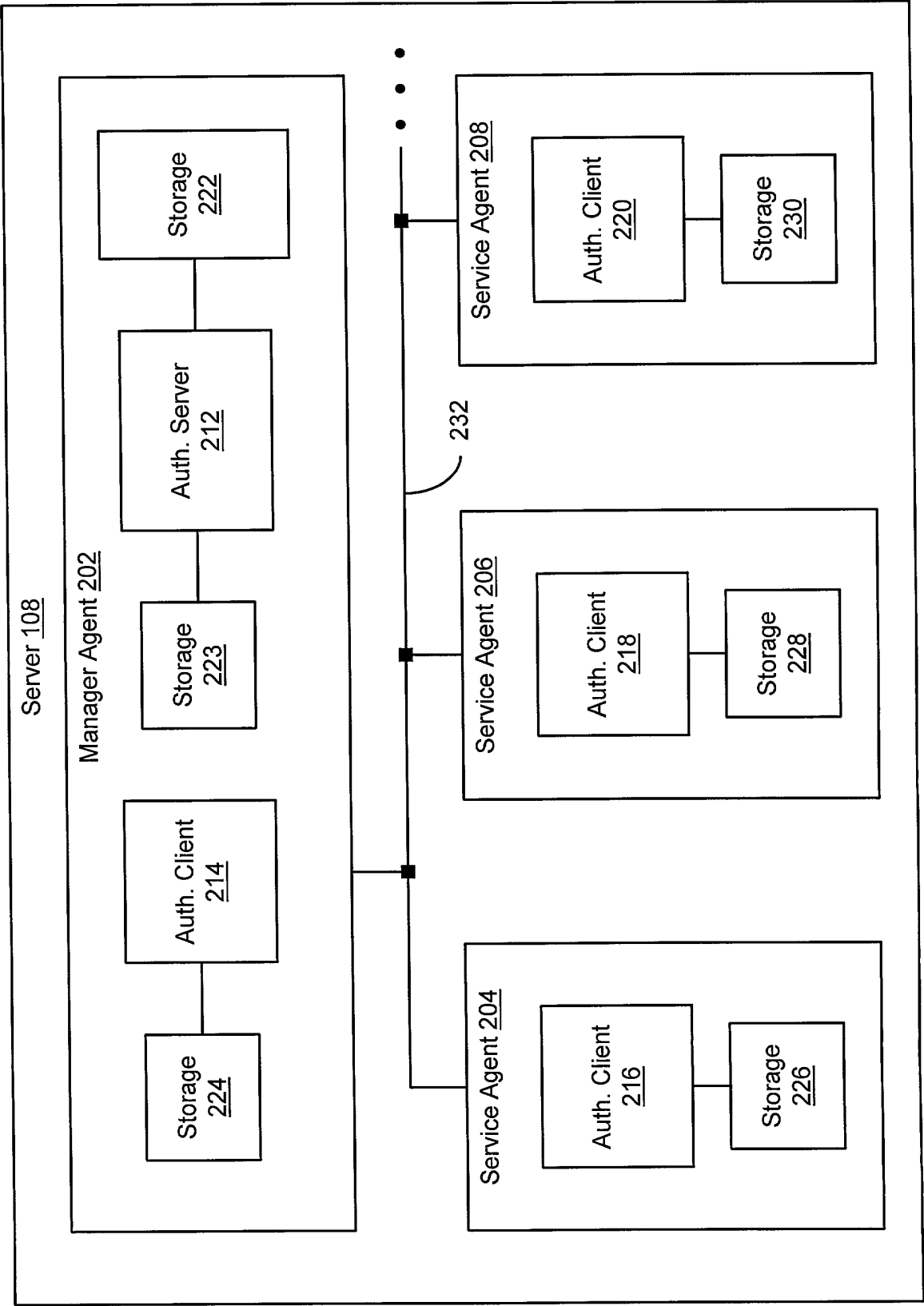


Fig. 2



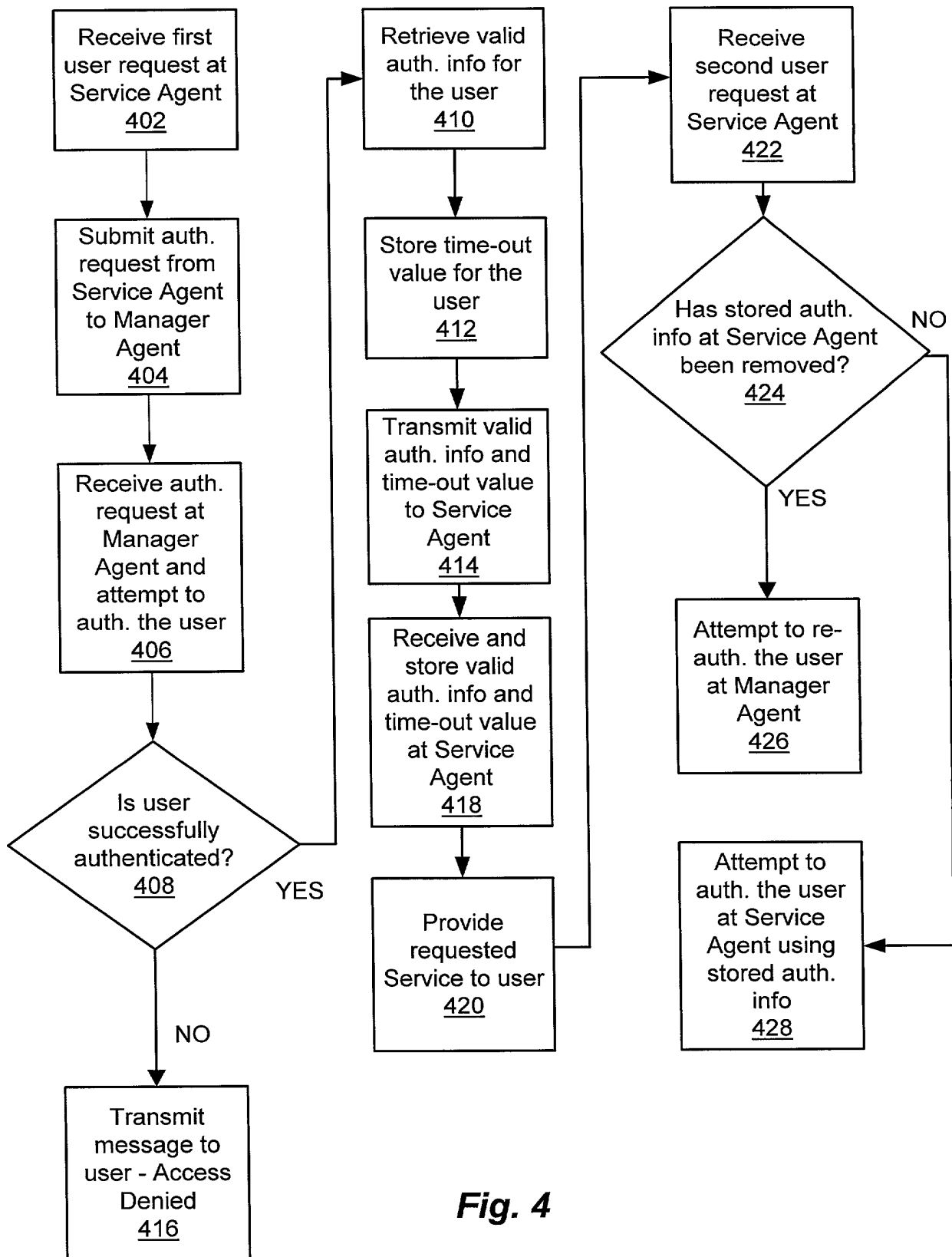


Fig. 4

DECLARATION AND POWER OF ATTORNEY

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: DISTRIBUTED SYSTEM AUTHENTICATION

the specification of which (check one):

☒ is attached hereto. ☐ was filed _____ as Application No. _____
amended on _____ (if applicable).

☐ was filed as PCT International Application No. _____ on _____,
and was amended under PCT Article 19 on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations §1.56(a).

I hereby claim foreign priority benefits under Title 35, USC §119(a)-(d) of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>	<u>Date Filed</u>	<u>Priority Claimed</u>	
_____ (Number) (Country)	_____ (Day/Month/Year)	<input type="checkbox"/> Yes	<input type="checkbox"/> No
_____ (Number) (Country)	_____ (Day/Month/Year)	<input type="checkbox"/> Yes	<input type="checkbox"/> No

I hereby claim the benefit under Title 35, USC §119(e) of any United States provisional application(s) listed below:

_____ (Application Number)	_____ (Filing Date)
_____ (Application Number)	_____ (Filing Date)
_____ (Application Number)	_____ (Filing Date)

Express Mail Number

EL418425315US

Attorney

Docket No.: SYNER-161XX

I hereby claim the benefit under Title 35 USC §120 of any United States application(s) listed below and insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35 USC §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application No.)	(Filing Date)	(Patented/pending/abandoned)
-------------------	---------------	------------------------------

(Application No.)	(Filing Date)	(Patented/pending/abandoned)
-------------------	---------------	------------------------------

(Application No.)	(Filing Date)	(Patented/pending/abandoned)
-------------------	---------------	------------------------------

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) to prosecute this application and transact all business connected therewith in the Patent and Trademark Office, and to file with the USRO any International Application based thereon.

Stanley M. Schurgin, Reg. No. 20,979
 Charles L. Gagnebin III, Reg. No. 25,467
 Paul J. Hayes, Reg. No. 28,307
 Victor B. Lebovici, Reg. No. 30,864

Eugene A. Feher, Reg. No. 33,171
 Beverly E. Hjorth, Reg. No. 32,033
 Holliday C. Heine, Reg. No. 34,346
 Gordon R. Moriarty, Reg. No. 38,973

Address all correspondence to:

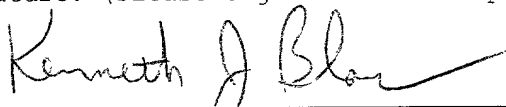
WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
 Ten Post Office Square
 Boston, Massachusetts 02109
 Telephone: (617) 542-2290
 Telecopier: (617) 451-0313

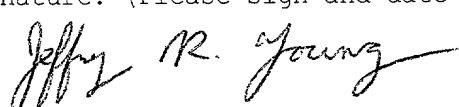
I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor: Smaragda Hadjinikitas		
City of Residence Somerville	State or Country Massachusetts	Country of Citizenship Canada
Post Office Address 38 Marion Street	City Somerville	State or Country Zip Code Massachusetts 02143
Signature: (Please sign and date in permanent ink.) X <i>Smaragda Hadjinikitas</i>		Date signed: X May 30 - 2000

Attorney

Docket No.: SYNER-161XX

Full Name of Second Joint Inventor: Kenneth J. Blanc		
City of Residence Natick	State or Country Massachusetts	Country of Citizenship USA
Post Office Address 142 Mill Street	City Natick	State or Country Zip Code Massachusetts 01760
Signature: (Please sign and date in permanent ink.) X 		Date signed: X 5/30/00

Full Name of Third Joint Inventor: Jeffrey R. Young		
City of Residence Upton	State or Country Massachusetts	Country of Citizenship USA
Post Office Address 48 South Street	City Upton	State or Country Zip Code Massachusetts 01568
Signature: (Please sign and date in permanent ink.) X 		Date signed: X 5/30/00